REMARKS

Claims 1-31 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(e) Rejection:

The Office Action rejected claims 1, 5, 7-10, 13-16, 18-21, 26, 27 and 31 under 35 U.S.C. § 102(e) as being anticipated by Schofield (U.S. Patent 6,263,485).

Schofield teaches a method for defining Interface Definition Language-defined data types, operations, or interfaces by generating an ASCII string descriptor that identifies the data type, interface, or operation (Schofield, Abstract). Specifically, Schofield teaches a code generator 112 operable to generate files in the language of the client and server applications, and to produce a Compact IDL Notation (CIN) (Schofield, col. 6, lines 23 – 51).

Contrary to the Examiner's assertion regarding claim 1, Schofield fails to teach a client generating a request *for* type information for an attribute or event, wherein the *request is expressed in an interface definition language.* Instead, Schofield teaches converting an original IDL description into new ASCII string descriptors that are "contained in a header file that is linked into both the client and server applications" (Schofield, column 3, lines 59-62, and column 11, lines 24-26). Further, Schofield teaches the benefits of using these ASCII strings instead of IDL to describe operations and data types (Schofield, column 4, lines 5-11). Thus, **Schofield teaches away** from using IDL to express requests. In addition, Schofield does not teach a client generating a request *for type information.* As the Examiner states in the Response to Arguments section of the Final Office Action, Schofield teaches that client requests *include* data type definitions so that a capsule can create and/or load an appropriate object according to a received request (Schofield, col. 6, lines 1-12). Applicants note that in the Response to Arguments section the Examiner asserts that Schofield teaches generating requests *for*

type information, but only cites references where Schofield describes requests *including* type information. These are two very different types of requests. A request that already *includes* type information is clearly not a request *for* the type information. In Schofield, the requester already has type information and includes that type information in client requests for object operations. In fact, central to Schofield's system is a method for compiling IDL source files that contain "interface definitions, operation definitions, and individual data type definitions" (Schofield, col. 7, lines 11-14) into client and server applications (See also, Schofield, col. 6, lines 13-49). Thus, Schofield clearly fails to teach a client generation a request ***for*** type information.

Applicants also respectfully disagree with the Examiner's characterization of code generator 112 as an object request broker. As discussed above, Schofield's code generator 112 is operable to generate files in the language of the client and server applications, and to produce a Compact IDL notation as illustrated in Figs. 5 – 7 and described starting at col. 7, line 20. Schofield does not teach that code generator 112 is sent requests for type information as asserted by the Examiner. Schofield is very clear that code generator 112 "parses a source file line by line" and "produces an ASCII descriptor in the header file to be included by the client and server applications" (Schofield, col. 6, lines 50-64).

Likewise, Applicants respectfully disagree with the Examiner's characterization of server stub 87 from Fig. 4 as a metadata gateway. Schofield teaches that "code generator 112 produces a client stub file 79 containing client stub functions and a server stub file 89 containing definitions for object implementations" (Schofield, col. 6, lines 30 – 32). Schofield does not teach that server stub 89 receives requests for type information from code generator 112, which the Examiner views as an object request broker. In fact, Schofield describes how code generator 112 creates server stub 89. Once Schofield's system is running (and requests are generated and sent) code generator 112 would not even be executing. Code generator 112 is only used to compile portions of client and server applications. Applicants fail to see how code generate 112 can be considered a metadata gateway that receives a request for type information from an object request

broker. **Although Applicants have previously asserted this argument, the Examiner has not supplied any rebuttal of this specific argument.**

Applicants further disagree with the Examiner's characterization of implementation library 81 from Fig. 4 as a metadata repository. Schofield teaches that the implementation library includes executable code such as the server applications and server stubs generated from the IDL specification of the object's interface (Schofield, column 5, lines 55-59). Schofield clearly describes how a compiler links the server application object code and the server stub object code to produce implementation library 81 (Schofield, col. 6, lines 47-49). Thus implementation library 81 contains server routines in executable format and thus cannot be a metadata repository from which type information is read and that stores the type information in a *database* format. **Although Applicants have previously asserted this argument, the Examiner has not supplied any rebuttal of this specific argument.**

Applicants submit that Schofield also fails to teach translating the type information from the database format to the interface definition language. Firstly, as shown above, Schofield does not teach storing type information in a database format, but rather compiles header files for inclusion in client and server applications. Secondly, Schofield describes generating a CIN (ASCII based) descriptor ***from*** an IDL data type (Schofield, column 7, lines 20-22). In his Response to Arguments, the Examiner incorrectly states Applicants' argument. Applicants actually argue (as previously presented) that Schofield fails to teach translating the type information ***from*** *the database format* ***into*** *the interface definition language.* Instead, Schofield describes his method for generating a CIN (ASCII based) descriptor ***from*** an IDL data type (Schofield, column 7, lines 20-22). In fact, Schofield clearly illustrates this with an example where an ADD operation is translated from its IDL descriptor, "long Add (in long x, in long y);" to its equivalent CIN descriptor, "126861413+3+ADDA3+DFAFAF0+0+" (Schofield, column 10, lines 41-51). Hence, **Schofield teaches the opposite** of translating the type information ***into*** the interface definition language.

Further, Applicants disagree with the Examiner's assertion that Schofield teaches "the client receiving the translated type information for the attribute or event through the object request broker, wherein the translated type information is expressed in the interface definition language." As shown above, Schofield fails to teach translating type information into IDL. Thus, any **translated type information** in Schofield is expressly taught as **not in IDL**, but in Schofield's ASCII based CIN descriptor language (See, Schofield, Abstract, col. 3, line 59 – col. 4, line 11, Fig. 5, and col. 7, line 20 – col. 11, line 36).

Applicants remind the Examiner that for a rejection under section 102, the ***identical*** invention must be shown in as complete detail as is contained in the claims. M.P.E.P 2131; *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, ***arranged as in the claim***. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). Applicants submit that the rejection of claim 1 is clearly not supported by the teachings of the cited art for at least the numerous reasons stated above and respectfully request withdrawal of the rejection. Similar arguments apply in regard to the rejection of independent claims 10, 14, 22 and 27.

Section 103(a) Rejection:

The Office Action rejected claims 2-4, 6, 11, 12, 17, 23-25 and 28-30 under 35 U.S.C. § 103(a) as being unpatentable over Schofield in view of Kulkarni et al. (U.S. Patent 5,848,243). Applicants respectfully submit that this rejection is improper for at least the reasons given above in regard to the rejection of the independent claims.

In regard to the rejections under both sections 102 & 103, Applicants also assert that the rejections of numerous ones of the dependent claims are further unsupported by the teachings of the cited art. However, since the rejection of the independent claims has

been shown to be improper, a further discussion of the rejection of the dependent claims is not necessary at this time.

Information Disclosure Statement:

Applicants note that an information disclosure statement was submitted electronically on November 12, 2003. Applicants again respectfully request the Examiner to carefully consider the listed references and return a copy of the signed and initialed electronic submission from this statement.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-46200/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

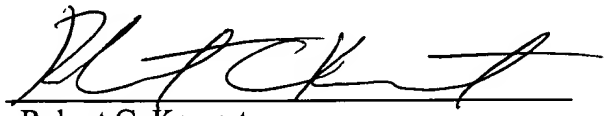
☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Fee Authorization Form authorizing a deposit account debit in the amount of $ for fees ( ).

☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: August 23, 2004